

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-338824

(43)Date of publication of application : 10.12.1999

(51)Int.Cl. G06F 15/00  
G06F 13/00

(21)Application number : 11-102757

(71)Applicant : HEWLETT PACKARD CO &lt;HP&gt;

(22)Date of filing : 09.04.1999

(72)Inventor : STEELE DOUGLAS W  
GOIN TODD M  
BRYANT CRAIG W

(30)Priority

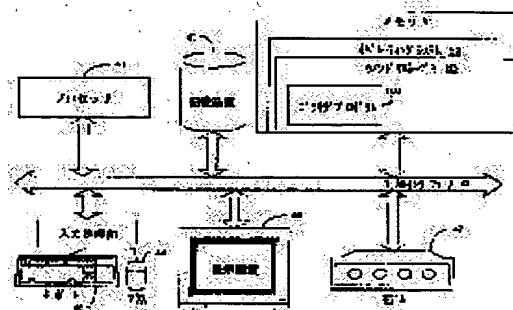
Priority number : 98 60875 Priority date : 15.04.1998 Priority country : US

## (54) METHOD FOR PROTECTING WEB RESOURCE

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To achieve flexible high-grade security in the case of accessing a web resource on a by a client system.

**SOLUTION:** In the client system, a memory 51 is provided with an operating system 52 and a window manager 53. A token is generated by a client browser 100 inside the memory 51 of a computer and transferred to the security server together with a service request. Besides, these service request and token are supplied to a server application, the verification of the token is requested, the received token is transferred to the security server and tokens from the client browser 100 and the server application are compared. In the case of coincidence in this comparison, a coincidence notice is generated but when these tokens are not coincident, a non-coincidence notice is generated and transferred to the server application. When the coincidence notice is received, the requested service is provided.



## LEGAL STATUS

[Date of request for examination] 19.12.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

(51)Int.Cl.<sup>6</sup>G 0 6 F 15/00  
13/00

識別記号

3 3 0  
3 5 4

F I

G 0 6 F 15/00  
13/003 3 0 A  
3 5 4 Z

## 審査請求 未請求 請求項の数1 O L (全 12 頁)

(21)出願番号 特願平11-102757

(22)出願日 平成11年(1999)4月9日

(31)優先権主張番号 0 9 / 0 6 0 - 8 7 5

(32)優先日 1998年4月15日

(33)優先権主張国 米国 (US)

(71)出願人 398038580

ヒューレット・パッカード・カンパニー  
HEWLETT-PACKARD COMPANY

アメリカ合衆国カリフォルニア州パロアルト ハノーバー・ストリート 3000

(72)発明者 ダグラス・ダブリュー・スティーレ  
アメリカ合衆国 コロラド、フォートコリーンズ、マックマリー 5224

(74)代理人 弁理士 萩野 平 (外4名)

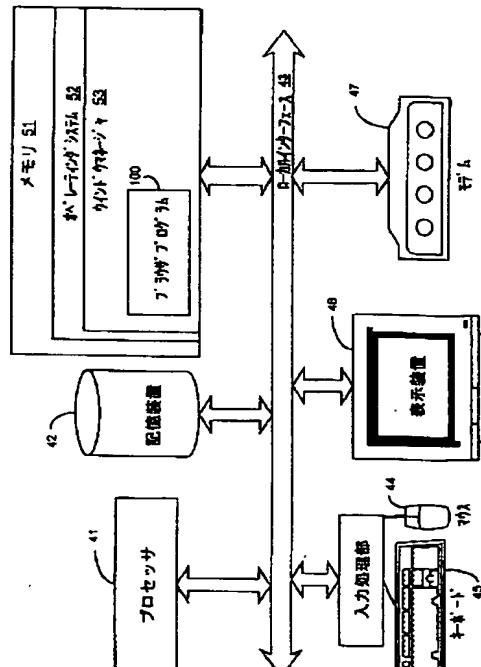
最終頁に続く

## (54)【発明の名称】 ウェブリソースの保護方法

## (57)【要約】

【課題】 クライアントシステムによって、サーバ上のウェブリソースにアクセスする際に、融通性のある、かつ高度なセキュリティ実施を可能にする。

【解決手段】 クライアントシステムにおいて、メモリ51は、オペレーティングシステム52とウインドウマネージャ53を備えている。コンピュータのメモリ51内のクライアントブラウザ100でトークンを生成し、セキュリティサーバに対してサービス要求とともに転送する。また、このサービス要求及びトークンをサーバアプリケーションに供給してトークンの検証を要求し、受信したトークンをセキュリティサーバに転送して、クライアントブラウザ100及びサーバアプリケーションからのトークンを比較する。この比較で一致した際に一致通知を生成し、一致しない場合に不一致通知を生成してサーバアプリケーションに転送する。一致通知を受信した際に要求されたサービスを提供する。



1

## 【特許請求の範囲】

【請求項1】 通信ネットワークシステムにおけるウェブリソースの保護方法であつて、  
 クライアントブラウザ(100)に対してトークンを生成するステップと、  
 セキュリティサーバ(140)に対して前記トークンを転送するステップと、  
 サービス要求及び前記トークンを転送するステップと、  
 前記クライアントブラウザ(100)からの前記サービス要求及び前記トークンを受信するアプリケーションサーバ(160)を提供するステップと、  
 前記アプリケーションサーバ(160)によって前記トークンの検証を要求するステップと、  
 前記アプリケーションサーバ(160)が受信した前記トークンを前記セキュリティサーバ(140)に転送するステップと、  
 前記クライアントブラウザ(100)から受信した前記トークンと前記アプリケーションサーバ(160)から受信した前記トークンとを比較するステップと、  
 を有することを特徴とするウェブリソースの保護方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、コンピュータ及びソフトウェアに関し、特に、クライアントブラウザ( client browser)によってサーバ上のウェブリソース( web resource)にアクセスする際に伴うセキュリティ(保護)を実施するウェブリソースの保護方法に関する。

## 【0002】

【従来の技術】 従来、インターネットは、通信プロトコルのTCP/IP (Transmission Control Protocol/Internet Protocol) を適用して相互通信を行う通信ネットワーク及びゲートウェー( gateway) を接続した世界的なコンピュータ・ネットワーク集合体である。インターネットは、その中心として、データ及びメッセージを伝送する膨大な商用、政府、教育用及びその他のコンピュータシステムからなる主ノード又はホストコンピュータ間を高速データ通信回線で接続している。

【0003】 WWW( World Wide Web) は、世界中のHTTP ( Hypertext Transfer Protocol) を用いたサーバ( 以後、HTTPサーバと記載する) にインターネットリンク( interlink) されたハイパーテキストの文書の全体を参照する。WWW上の文書は、ページ又はウェブ( Web) ページと呼称され、インターネット上のロケーション( location) を示すURL( Uniform Resource Locator) で識別されるHTML( HyperText Markup Language) によって記載されている。このURLは、特定のマシン及びパス名を指定し、このURLによって、HTTPに基づいたファイルへのアクセス、及び、ファイルのノード間からエンドユーザへの伝送を可

10

20

30

40

50

2

能している。ウェブサイト( Web site) は、これらの文書と、WWW上のHTTPサーバによって提供される関連ファイル、スクリプト( script) 、サブプロシージャ( subprocedure) 及びデータベースとの関連グループである。

【0004】 ユーザは、ウェブサイトにアクセスするために、ブラウザプログラム( browser program) とインターネット接続とを必要とする。ブラウザプログラムは、「ウェブブラウザ( web browser) 」とも呼ばれ、ユーザが、インターネットをナビゲート( navigate) し、WWW、他のネットワーク又はユーザのコンピュータ上でHTML文書を閲覧できるようにするためのクライアントアプリケーション( client application) である。また、ブラウザプログラムによってユーザは、HTML文書内に組込まれている「タグ( tag) 」と呼ばれるコードを指定した処理が実行できる。このタグは、文書内の特定の語及び画像をURLと関連付けることによって、ユーザが、キーを押下したとき、又は、マウスをクリックしたときに、世界中のどこかにある他のファイルにアクセスできるようにするためのものである。

【0005】 これらのファイルには、Java applet、perl application、他のスクリプト言語( scripted language) 、active X-control、又は、ユーザがリンク上でクリックすることによって起動すると実行する他の小容量の内蔵型ソフトウェアプログラムと同様に、テキスト( あらゆるフォント及びスタイルのもの) と、グラフィック画像と、映像ファイルと、サウンドとが含まれる。スクリプトは、クライアントユーザ( client user) による要求に応じてHTTPサーバが実行するアプリケーションである。これらのスクリプトは、単一ジョブを行うHTTPデーモン( HTTP daemon) によって呼ばされ、その後終了する( exit) 。

【0006】 スクリプトの一つのタイプとして、共通ゲートウェイインターフェース( common gateway interface, CGI) スクリプトが周知である。CGIスクリプトは、一般的にユーザがリンク又は画像などのウェブページ内の要素をクリックした際に、呼び出される。CGIスクリプトを使用することによって、ウェブページ内の対話性が得られる。このCGIスクリプトは、C、C++、Perlのような多くの言語で記載できる。また、CGI-BINとは、HTTPサーバが実行することが出来るCGIスクリプトのアプリケーションのライブラリ( library) である。

【0007】 WWW上のHTTPサーバによって提供されるこれらの文書及び関連ファイル、スクリプト、サブプロシージャ及びデータベースに対してアクセスする場合、セキュリティが重要な問題となる。すなわち、許可されたクライアントシステムからの許可されたユーザのみに対して、HTTPサーバのアプリケーションへのアクセスを許可するように、その検証をどのようにする

か、また、悪意のある目的でアクセスを不正使用できないようにする保証をどのようにするかの課題がある。

【 0 0 0 8 】この課題に対して、現在使用されている方法に、「Cookie」を使用したものがある。Cookieは、サーバが、クライアントからの要求に応じてクライアントに返送するデータのブロック（以後、データブロックと記載する）である。このデータブロックは、クライアントシステムに格納されている。クライアントは、同一のウェブサイトに戻るときに、Cookieのコピーをサーバに送信し、この送信によってサーバがクライアントを識別する。Cookieは、ユーザを識別したり、サーバに対して要求されたウェブページのカスタマイズされたバージョンを送信するよう命令したり、ユーザにアカウント情報を提示したりする場合の管理目的に使用される。ほとんどのシステムでは、Cookieプログラムはユーザのログオン中に稼動する。

【 0 0 0 9 】このようなウェブリソースにアクセスする際にセキュリティを提供する従来の方法には、以下のようなセキュリティに関する欠点がある。本発明がいかにこれららの課題に取り組んで解決しているかについては、この後で説明する。

#### 【 0 0 1 0 】

【 発明が解決しようとする課題】従来の解決法の問題は、ホストアドレス及びユーザ名（すなわち、ユーザのログオン情報）が、非常に「スプーフィング（ spoofing ）」を受け易い普通テキストで送信されていることである。それゆえ、精通したハッカーは、他のマシン又は他のユーザからのように見せかけたパケット伝送によって無許可でサーバにアクセスすることができる。

【 0 0 1 1 】また、複数レベルのユーザのセキュリティ<sup>30</sup>が試みられている場合、他の問題が発生する。すなわち、Cookieを用いた方法では、セキュリティの単一レベルにのみ対応が可能である。更に、Cookieを使用する場合、ユーザアプリケーションはセキュリティシステムに統合することが出来ない。現在、Cookieはクライアントのブラウザプログラムの一部であり、ユーザアプリケーションからは独立している。

【 0 0 1 2 】従来技術における他の問題は、セキュリティに対する検証（ authentication ）が的確に出来ないことである。これは、サーバが、証明（ proof ）無しに、伝送中に識別されるユーザ名及びホスト名を受け入れるからである。更に、各コマンドのトランザクション（ transaction ）が独立しているため、いかなる状態も維持されないという問題がある。これによって、これらの方法が、「リプレイアタック（ replay attack ）」（ハッカーが正当なネットワークパケットを獲得し、いくつかの詳細（ユーザ名又は実行コマンド等）を変更して再度送信する）を受け易くなる。

【 0 0 1 3 】このように、上記従来例では、今までネットワークシステムは、インターネット又は他の種類の

ネットワーク上のウェブ文書（又ウェブリソースは）に対して、融通性（ flexible ）を備え、かつ、高度なセキュリティの実施を提供できないとい欠点があった。

【 0 0 1 4 】本発明は、このような従来の技術における課題を解決するものであり、クライアントブラウザによって、サーバ上のウェブリソースにアクセスする際に、融通性のある、かつ高度なセキュリティの実施が可能になるウェブリソースの保護方法の提供を目的とする。

【 0 0 1 5 】本発明のある目的、利点及び新規な特徴について、一部は以下で説明するが、この一部は以下の内容から、その調べによって当業者で明らかになるものである。あるいは、本発明を実施することによって知ることが出来るものである。本発明の目的及び利点は、特に特許請求の範囲で示す手段及び組合せによって、実現できるものである。

#### 【 0 0 1 6 】

【 課題を解決するための手段】上記課題を達成するため、本発明の一実施形態によれば、クライアントシステムのクライアントユーザインタフェース（ ブラウザプログラム ）がトークン（ token ）を生成する。このトークンは、セキュリティサーバ（ security sever ）へ送信されて、クライアントユーザのサービス要求に対する第三者による妥当性の検証（ third party validation ）を行う。そして、クライアントユーザインタフェースは、サービスのためにサーバアプリケーション（ sever application ）を呼び出し、サービスのために呼び出しの引数（ argument ）としてトークンをサーバアプリケーションに対して送信する。

【 0 0 1 7 】サーバアプリケーションは、クライアントユーザインタフェースからのサービス要求を受信し、その後クライアントユーザについてそれ自身のログイン検証プログラム（ login authorization program ）を実行する。ここで検証の結果がOKである、すなわちログインが正当である場合、サーバアプリケーションは、要求されたサービスについて必要なCGI-BINアプリケーションプログラム（ 以後、CGI-BINプログラムと記載する ）に対する呼び出しを実行する。

【 0 0 1 8 】要求されたサービスのために呼出されたCGI-BINプログラムは、プログラム名等のサービス識別子（ service indicator ）及びクライアントユーザインタフェースが生成したトークンが含まれる引数を受信する。要求されたプログラムは、セキュリティサーバへの接続を確立し、その後、引数として受信したトークンを、検証のためにセキュリティサーバへ送信する。

【 0 0 1 9 】セキュリティサーバは、要求されたプログラムから検証するトークンを受信し、要求されたプログラムを実行して受信したトークンを、クライアントユーザインタフェースを通じて受信したトークンを用いて検証する。セキュリティサーバは、この二つのトークンが一致した場合に、要求されたプログラムに対し、そのト

5

一クンが照合( *verify* ) したとの指示を返送する。セキュリティサーバは、要求されたプログラムについてトークンを検証した後に、直ちにクライアントユーザインターフェースからトークンを受信する待ち状態に戻る。

【 0020 】CGI-BINプログラムは、要求されたプログラムを実行し、その終了前にサーバアプリケーションに対して出力を送信する。サーバアプリケーションは、要求されたプログラムの出力を受信し、クライアントユーザに表示するためにクライアントユーザインターフェース( ブラウザプログラム) に、そのデータを返送する。この時点で、サーバアプリケーションは、クライアントユーザインターフェースからのサービス要求を待つ状態に戻る。

【 0021 】本発明の他の実施形態によれば、ウェブリソースの保護のために、複数レベルのユーザのセキュリティを実施する。

【 0022 】本発明の他の実施形態によれば、ウェブリソースの保護を実現する装置及び方法では、ユーザアプリケーションに対し、統合されたセキュリティシステムを提供する。

【 0023 】このような本発明のウェブリソースの保護方法は、クライアントブラウザによって、サーバ上のウェブリソースにアクセスする際に、融通性のある、かつ高度なセキュリティの実施が出来るようになる。

【 0024 】本明細書に組込まれその一部を形成する添付図面は、本発明のいくつかの様相を説明しており、明細書中の説明とともに、本発明の原理を説明している。

【 0025 】

【 発明の実施の形態】本発明は、これらの図面に関連して説明するが、その中で開示した実施の形態に限定する意図はない。この反対に、本発明は、特許請求の範囲で定義する本発明の技術思想及び範囲内に含まれる全ての代替物、変形例及び均等物を全て包含することを意図している。

【 0026 】次に、本発明のウェブリソースの保護方法の実施の形態を図面を参照して詳細に説明する。

【 0027 】図1は、本実施形態に係るインターネットを利用するクライアント／サーバシステムの一構成を示す構成図であり、本発明の融通性と、拡張性と、プラットフォーム( platform ) の独立性とを説明している。このようなシステム構成は多くの形態で実現できるが、図1では、ネットワーク18、例えば、ローカルエリアネットワーク( LAN 、なお、LANに限定されない)に直接に接続されたUNI X又はPC等の複数の異なるワークステーション12、16を有している。追加のラップトップ型のワークステーション21、22は、同様に遠隔地に配置されており、ダイヤルイン( dial-in ) 又は他のネットワーク接続処理手順( 通信プロトコル ) 24によってネットワーク18と通信を行うようになっている。図1では、クライアント装置に該当する各

6

ワークステーション12、16、21、22が異なったハードウェアプラットフォームで構成しても良いことを強調するために、各ワークステーション12、16、21、22を一意的に図示した。

【 0028 】周知のように、ブラウザアプリケーション( browser application ) は、種々のハードウェアのプラットフォームに提供され、容易な使用が可能である。ブラウザは、ブラウザプログラムを用いて、インターネット32によって情報にアクセスするためのユーティリティ( utility ) を有していることで最も一般的に認められている。このブラウザは、ユーザがあらゆるサービス集団と考えることが出来る装置又はプラットフォームであり、HTTPを使用して、ウェブサーバ31又はネットワークサーバ26から情報を検索し、HTMLコードを解釈し、フォーマットして、その翻訳結果をワークステーションの表示装置で表示する。

【 0029 】追加のワークステーション33、34は、同様に配置されており、ローカルサーバ及びインターネット32上のウェブページにアクセスするために、ウェブサーバ31と通信を行うようになっている。ワークステーション33、34は、LAN35上のウェブサーバ31との間での通信を行う。ネットワーク18及びLAN35は、例えば、10BASE2、10BAS5、10BASF、10BAST、BASE BANネットワーク、及び、CO-EXケーブル等として周知のイーサネット型ネットワークであっても良い。なお、以下、上記ワークステーション12、16、21、22、33、34は、クライアントシステムと記載し、ネットワークサーバ26及びウェブサーバ31とともにサーバシステムと記載する。また、ネットワーク18及びLAN35をネットワーク18、35と記載する。

【 0030 】図2は、実施形態にあってクライアントシステムのコンピュータシステム内のブラウザプログラム100を示すロック図であり、ワークステーション12、16、21、22、33、34のクライアントシステムのアーキテクチャを示している。図2に示すように、一般的に、クライアントシステムには、ネットワーク11上のロケーション( location ) へのアクセス用としてクライアントブラウザのブラウザプログラム100( 例えば、ネットスケープ( Netscape ) 、インターネットエクスプローラ( Internet Explorer ) 、又は他のブラウザプログラム等) のみが用いられている。これらのブラウザプログラム100は、メモリ51内に格納され、かつ、通信機能のモジュール47からアクセスして、ネットワーク11に接続された他のウェブリソースをユーザに移送( トランスポート( transport ) ) する。ウェブリソースを見つけるために、ユーザは、URLによって示されるウェブリソースのネットワーク11上のロケーションを知らなければならない。これらのURLは、しばしば暗号のようであり、それらの名前付けの規則に

30

40

40

50

における非常に複雑な体系及びフォーマットに基づいてい

る。

【 0031 】このようなクライアントシステムでは、プロセッサ41と、記憶装置42と、オペレーティングシステム52及びウインドウマネージャ53を備えたメモリ51とを使用して、ユーザが所望するウェブリソースを識別し、かつ、アクセスして処理する。プロセッサ41は、バス(ローカルインターフェース)43を通じてメモリ51及び記憶装置42からデータを受信する。ユーザからの操作指示は、入力装置のマウス44及びキーボード45を使用して信号化される。動作入力及び結果出力は、表示装置46で表示される。

【 0032 】図2に示すように、本実施形態では、ウインドウマネージャ53にブラウザプログラム100が設けられている。ブラウザプログラム100は、サーバと対話して、クライアントユーザが要求したデータ得るとともに、機能処理を実行するソフトウェアである。このブラウザプログラム100の詳細については、以降の図4及び図5を参照して説明する。

【 0033 】図3は、実施形態にあってサーバシステムのコンピュータシステム内のサーバアプリケーションプログラム(以後、サーバアプリケーションと記載する)120とCGI-BINアプリケーションプログラム160(以後、単にCGI-BINプログラムという。)とセキュリティサーバ140とを示すブロック図であり、ネットワークサーバ26及びウェブサーバ31(図1参照)のサーバシステムでのアーキテクチャを示している。図1に示すサーバシステム26, 31とクライアントシステム12, 16, 21, 22, 33, 34との主な違いは、クライアントシステムが、クライアントユーザにインタフェースし、ブラウザプログラム100によって機能性を要求するのに対し、サーバシステム26, 31は、サーバアプリケーション120と、セキュリティサーバ140と、アプリケーションサーバに該当するCGI-BINプログラム160とを利用して、クライアントが要求するサービスを提供する点が相違する。

【 0034 】この相違する以外では、サーバシステムは、プロセッサ61と、記憶装置62と、マウス64と、キーボード65と、表示装置66と、モデム67との機能が、基本的に前記した図2のクライアントシステムの対応する要素と同一である。本技術分野で周知のように、クライアントシステム12, 16, 21, 22, 33, 34及びサーバシステム26, 31は、同様な物理的なマシン上にある。

【 0035 】サーバシステムにおける主な相違は、オペレーティングシステム72及びウインドウマネージャ73と対話するためのメモリ71が、HTTPデーモン(以後、HTTPDと記載する)によって呼び出されるサーバアプリケーション120と、CGI-BINプロ

グラム160と、セキュリティサーバ140とを利用して、クライアントが要求するサービスを提供する点である。このサーバアプリケーション120と、CGI-BINプログラム160と、セキュリティサーバ140との詳細については、以降の図4、図6、図7及び図8を参照して説明する。

【 0036 】図4は、クライアントユーザインターフェース(ブラウザプログラム)100と、サーバアプリケーション120と、セキュリティサーバ140と、CGI-BINプログラム160とのプロセスを示すブロック図である。図4に示すように、クライアントシステム12, 16, 21, 22, 33, 34は、ブラウザプログラムのクライアントユーザインターフェース100を利用することによって、サーバシステム31にサービスを要求することが出来る。

【 0037 】ブラウザプログラムは、まず、ユーザから要求を受け取って、ユーザ名及びパスワードなどを検証して、そのユーザが特定の機能にアクセスすることが許可されているかを確かめるチェックを行う。

【 0038 】次に、ブラウザプログラム100は、適切なアルゴリズム及びジェネレータを利用してトークンを生成する。好ましい実施形態では、トークンは連続した数値ではないものであり、実際には乱数生成プログラムによって生成された数値である。

【 0039 】そして、クライアントユーザインターフェースは、セキュリティサーバ140に接続する。この接続は、例えば、ソケット(socket)を使用して行われる。ブラウザプログラム100は、確立した接続、例えばソケット接続を利用して、セキュリティサーバ140にトークンを送信する。次いで、ブラウザプログラム100は、サービスを要求するためにサーバアプリケーション120を呼び出し、要求されたサービスのための引数(argument)の一つとしてサーバアプリケーション120にトークンを送信する。このサービス要求は、ネットワーク回線を出てサーバシステム31に送出され、サーバアプリケーション120に受信される。

【 0040 】サーバアプリケーション120は、ブラウザプログラム100からのサービス要求を受信する。次に、サーバアプリケーション120は、要求されたプログラムを調べ、サービスのためのプログラム名及び実行引数(execute argument)を用いて、CGI-BINプログラム160を呼び出す(invoke)ことによって要求されたプログラムを呼び出す(call)。

【 0041 】CGI-BINプログラム160は、このプログラム名及び実行引数を受信する。要求されたサービスを提供する要求されたサブルーチンの実行前に、CGI-BINプログラム160が、セキュリティサーバ140とのソケットを確立する。一旦セキュリティサーバ140とのソケットが確立すると、CGI-BINプログラム160は、セキュリティサーバ140にトーク

9

ンを検証するためのトークン検証要求を送信する。

【0042】セキュリティサーバ140は、初期化されると、受信ソケット（listen socket）を確立する。セキュリティサーバ140は、ブラウザプログラム100でユーザーのアクセスが検証され許可されたときに生成された接続上に確立されているソケットでブラウザプログラム100からのトークンを受信するのを待つ状態となる。トークンは、一旦クライアントユーザインタフェースから受信されると、セキュリティサーバ140のトークン検証テーブル（token verification table）に加えられる。セキュリティサーバ140は、CGI-BINプログラムとのトークンを検証するために確立されたソケットでCGI-BINプログラム160からのトークン検証要求を受信するのを待つ状態となる。トークン検証要求がCGI-BINプログラム160から受信されると、セキュリティサーバ140は、トークン検証テーブルをチェックし、そのトークンがクライアントユーザインタフェースから受信したものであり、したがって、正当なトークンであるか否かについて、CGI-BINプログラム160に対してトークン検証メッセージを返送する。

【0043】CGI-BINプログラム160は、セキュリティサーバ140からトークン検証メッセージを受信すると、トークンの検証をチェックする。セキュリティサーバ140から受信したトークン検証メッセージが満足できるもの（OK）である場合、すなわちトークンが正当なものである場合、CGI-BINプログラム160は、要求されたプログラムを実行し、標準出力（std out）に出力を書込む。次に、標準出力は、その出力をサーバアプリケーション120に返送する。セキュリティサーバ140から受信したトークン検証メッセージが許容できないものである場合、すなわちトークンが正当でない場合、エラーメッセージがサーバアプリケーション120に送信される。出力がサーバアプリケーション120に送信されると、CGI-BINプログラム160が終了して存在しなくなる。

【0044】サーバアプリケーション120は、CGI-BINプログラム160の出力とCGI-BINプログラム160のプロセスの終了ステータスとを受信し、ブラウザプログラム100に対し、ネットワークを通じて出力を返送する。そして、ブラウザプログラム100は、クライアントシステム12においてサービスを要求したアプリケーションプログラムに対して出力を返送する。このプロセスについては、更に、図5から図9を参照して以下に説明する。

【0045】図5は、図4に示すクライアントシステムのブラウザプログラムが実行するプロセスを示すフローチャートである。図4及び図5を参照して、ブラウザプログラム100の最初のステップでは、ステップS101において、ブラウザプログラム100が初期化され

10

る。そして、ブラウザプログラム100は、ステップS102において、ユーザからユーザ名及びパスワードのログインを受け入れ、セキュリティサーバ140に接続する。次いで、ステップS103において、ユーザからサービス要求を受信する。

【0046】次いで、ブラウザプログラム100は、ステップS104においてトークンを生成する。好ましい実施形態では、トークンは乱数生成機能によって生成される乱数である。しかしながら、利用することが出来る一意的なトークンを生成する方法が他にもあることは、本技術分野において周知である。

【0047】そして、ブラウザプログラム100は、ステップS105において、ステップS104で生成されたトークンをセキュリティサーバ140に送信する。次いで、ステップS106において、サーバアプリケーション120に結合（又は接続）する。そして、ステップS107において、サーバアプリケーション120に対する呼び出しを行い、サーバアプリケーション120に対してトークンを引数のデータとして送信する。その後、ステップS108においてデータが返送されるまで処理を中断する。

【0048】データがクライアントユーザインタフェースに戻ってくると、ブラウザプログラム100は、ステップS108において再開し、ステップS109において、サーバアプリケーション120から受信した（出力された）データをユーザに対して表示する。そして、ステップS110のループを通じてステップS103に戻り、ユーザからの次のサービス要求を待つ状態になる。

【0049】図6は、図4に示すサーバシステムのサーバアプリケーションが実行するプロセスを示すフローチャートである。サーバアプリケーション120は、ステップS121において初期化される。そして、ステップS122において、クライアントシステムからのサービス要求を受信するのを待つ状態となる。

【0050】ステップS122においてサービス要求を受信すると、サーバアプリケーション120は、どのブラウザプログラム100がクライアントシステムが要求したサービスを提供するかを判断し、ステップS123において、要求された特定のCGI-BINプログラム160に結合する。サーバアプリケーション120は、ステップS124において、プログラム名及び実行引数等の特定の引数（その中の一つはトークンである）で特定のCGI-BINプログラムを呼び出し、必要なデータを送信する。サーバアプリケーション120は、ステップS125において、特定のCGI-BINプログラム160からデータを受信するまで処理を中断する。

【0051】サーバアプリケーション120は、ステップS126において、指定されたCGI-BINプログラム160からの出力を受信する。そして、ステップS127において、CGI-BINプログラム160から

50

11

受信した出力を書き込み、またサービスを要求しているクライアントシステムに対してその出力を返送する。その後、ステップS128において、そのセッションを終了し、ステップS122に折返し、新たなサービス要求を受信するまで処理を中断する。

【0052】図7は、図4に示すサーバシステムのセキュリティサーバ140のプロセスを示すフローチャートである。まず、セキュリティサーバ140が、ステップS141において初期化される。次に、ステップS142において、ユーザのログイン名及びパスワードを得ることによって、ブラウザプログラム100からの接続（例えはソケット接続）を受け入れる。そして、ユーザのログイン名及びパスワードを認証する。一旦ログインユーザ名及びパスワードの認証が完了すると、セキュリティサーバ140は、ステップS143において、ソケット接続でクライアントユーザインターフェース100からのトークンを受信するまで処理を中断する。

【0053】セキュリティサーバ140は、ステップS144において、CGI-BINプログラム160からソケット接続を受け入れると、ステップS145において、CGI-BINプログラムとのソケット接続でCGI-BINプログラム160からのトークン検証要求を受信する。次いで、ステップS143においてクライアントユーザインターフェースからソケット接続で受信したトークンを用いて、CGI-BINプログラム160からCGI-BINプログラムのソケット接続で受信したトークンを検証する。

【0054】ステップS143において受信したトークンが、ステップS145において受信したトークンと一致する場合は、そのトークンの検証は合格となる。ステップS143においてクライアントユーザインターフェースからソケット接続で受信したトークンが、ステップS145においてCGI-BINプログラム160から受信したトークンと一致しない場合、そのトークンの検証は不合格となる。セキュリティサーバ140は、所定の期間、タイムアウトする前にトークン検証要求がCGI-BINプログラム160から到着するのを待つ状態となる。トークンについて、タイムアウトしている場合にトークン検証要求がCGI-BINプログラム160から続けて送られてくると、トークンの検証が不合格となる。

【0055】セキュリティサーバ140は、ステップS147において、トークン検証の結果のトークン検証メッセージを、CGI-BINプログラム160に送信し、ステップS144において生成されたCGI-BINプログラムのソケット接続を閉じる。そして、ステップS148のループを通じてステップS143に戻り、クライアントユーザインターフェースから次のトークンを受信するまで待ち状態になる。

【0056】図8は、図4に示すCGI-BINプログ

12

ラムプロセスのプロセスを示すフローチャートである。まず、CGI-BINプログラム160は、ステップS161において初期化される。そして、ステップS162において、プログラム名及び引数によって要求されたサービスに関するサービス要求を受信する。次いで、ステップS163において、セキュリティサーバ140に対するソケットを確立する。好ましい実施形態では、TCP/IPソケットが確立される。

【0057】ステップS164において、サーバアプリケーション120から受信したトークンを、検証のためにセキュリティサーバ140に送信する。そして、ステップS165において、セキュリティサーバ140からトークン検証メッセージが返送されるまで処理を中断する。

【0058】一旦セキュリティサーバ140からトークン検証メッセージを受信すると、ステップS166において、トークンの検証に関するテストを実行する。セキュリティサーバ140によってトークンが照合されると、プロセスはステップS167に進み、CGI-BINプログラム160が、要求されたサービスのプログラムを実行する。ステップS167において要求されたサービスのプログラムが実行された後に、ステップS168において、要求されたサービスのプログラムから標準出力及び標準エラーメッセージを受信する。CGI-BINプログラム160は、ステップS169において、サーバアプリケーション120に対して標準出力及び標準エラーデータを送信し、その後ステップS172において終了する。

【0059】ステップS166におけるトークンの検証をチェックした結果、トークンが照合しない場合、ステップS171において、CGI-BINプログラム160は、サーバアプリケーション120に対し、セキュリティサーバ140によるトークン検証が失敗したことを示すエラーメッセージを送信する。その後、CGI-BINプログラム160は、ステップS172において実行を終了する。

【0060】他の実施形態では、CGI-BINプログラム160が、セキュリティサーバ140に対し、実行中のコマンドのセキュリティレベルをトークンとともに送信する。セキュリティサーバ160は、トークンを検証するとともに、クライアントユーザインターフェース（ブラウザ）100のセキュリティレベルをチェックする。セキュリティサーバ140が正当なクライアントユーザインターフェース（ブラウザ）100をチェックしていることを保証するために、トークンを乱数でユーザインターフェースのセキュリティサーバ140に対する接続のポート数をえたものとしても良い。クライアントユーザインターフェース（ブラウザ）100のセキュリティレベルは、セキュリティサーバ140が、セキュリティサーバ140への初期接続時においてクライアント

50

13

ユーザインターフェース( ブラウザ ) 100 を認証するときに決定される。

【 0061 】 前記した説明は、例示し、かつ、説明のために提示したものである。これは、全てを網羅することを意図したものではなく、本発明を開示したそのままの形態に限定することも意図していない。前記した教示内容を考慮して、明らかな変形例及び変更例が可能である。説明した実施の形態は、本発明の原理及びその実際的な適用例を最も良く例示するものとして選択し、その説明を行っており、これによって、当業者が、あらゆる実施の形態及び意図する特定の用途に適するよう にあらゆる改良を加えることが可能である。すなわち、本発明を利用することが出来る。このような変形例及び変更例は全て、特許請求の範囲が公正にかつ合法的に権利を与えられる範囲に基づいて解釈されるときに、それら特許請求の範囲によって決定される発明の範囲内にある。

【 0062 】 以下に本発明の実施の形態を要約する。

1. 通信ネットワークシステムにおけるウェブリソースの保護方法であって、クライアントブラウザ( 100 )に対してトークンを生成するステップと、セキュリティサーバ( 140 )に対して前記トークンを転送するステップと、サービス要求及び前記トークンを転送するステップと、前記クライアントブラウザ( 100 )からの前記サービス要求及び前記トークンを受信するアプリケーションサーバ( 160 )を提供するステップと、前記クライアントブラウザ( 100 )によって前記トークンの検証を要求するステップと、前記アプリケーションサーバ( 160 )が受信した前記トークンを前記セキュリティサーバ( 140 )に転送するステップと、前記クライアントブラウザ( 100 )から受信した前記トークンと前記アプリケーションサーバ( 160 )から受信した前記トークンとを比較するステップと、を有するウェブリソースの保護方法。

【 0063 】 2. 前記クライアントブラウザ( 100 )から受信したトークンと前記アプリケーションサーバ( 160 )から受信したトークンとが一致する場合に、一致通知を生成し、前記クライアントブラウザ( 100 )から受信したトークンと前記アプリケーションサーバ( 160 )から受信したトークンとが一致しない場合に、不一致通知を生成するステップを更に有する上記1記載のウェブリソースの保護方法。

【 0064 】 3. 前記生成された通知を、前記アプリケーションサーバ( 160 )に転送するステップを更に有する上記2記載のウェブリソースの保護方法。

【 0065 】 4. 前記一致通知を受信したときに、前記要求されたサービスを提供するステップを更に有する上記3記載のウェブリソースの保護方法。

【 0066 】 5. 前記トークンを生成するステップに、乱数生成プログラムを利用して前記トークンを生成するステップを更に有する上記1記載のウェブリソースの保

14

護方法。

【 0067 】 6. 通信ネットワークシステムにおいてウェブリソースを保護するコンピュータシステムにおいて、トークンを生成するクライアント装置( 12 )と、サービスを提供するアプリケーション装置( 160 )と、前記クライアント装置で生成された前記トークンを検証するセキュリティ装置( 140 )とを備えるコンピュータシステム。

【 0068 】 7. 前記トークンをサービス要求とともに前記アプリケーション装置( 160 )に転送する第1のクライアント機構と、第三者による妥当性を検証するために前記トークンを前記セキュリティ装置のセキュリティサーバ( 140 )に転送する第2のクライアント機構とを更に備える上記6記載のコンピュータシステム。

【 0069 】 8. クライアントブラウザ( 100 )からサービス要求及びトークンを受信する第1のアプリケーション機構と、前記セキュリティ装置( 140 )に対して前記トークンの検証を要求する第2のアプリケーション機構と、前記トークンの検証に関する一致通知を受信した際に、前記サービス要求で要求されたサービスを提供する第3のアプリケーション機構とを更に備える上記6記載のコンピュータシステム。

【 0070 】 9. クライアントブラウザ( 100 )からトークンを受信する第1のセキュリティ機構と、前記アプリケーション装置のアプリケーションサーバ( 160 )からのトークンを受信する第2のセキュリティ機構と、前記クライアントブラウザ( 100 )から受信したトークンと前記アプリケーションサーバ( 160 )から受信したトークンとを比較する第3のセキュリティ機構と、を更に備える上記6記載のコンピュータシステム。

【 0071 】 10. 前記クライアントブラウザ( 100 )から受信したトークンと前記アプリケーションサーバ( 160 )から受信したトークンとが一致した際に、一致通知を生成し、かつ、一致しない場合に、不一致通知を生成する第4のセキュリティ機構と、前記生成された通知を前記アプリケーションサーバ( 160 )に転送する第5のセキュリティ機構と、を更に備える上記9記載のコンピュータシステム。

【 0072 】

【 発明の効果】 以上の説明から明らかなように、本発明のウェブリソースの保護方法によれば、クライアントシステムによって、サーバ上のウェブリソースにアクセスする際に、融通性のある、かつ高度なセキュリティの実施が出来るようになる。

【 図面の簡単な説明】

【 図1】 本実施形態に係るインターネットを利用するクライアント/サーバシステムの一構成を示す構成図である。

【 図2】 実施形態にあってクライアントシステムのコンピュータシステム内のブラウザプログラムを示すプロッ

ク図である。

【図3】実施形態にあってサーバシステムのコンピュータシステム内のサービスアプリケーションプログラムとCGI-BINプログラムとセキュリティサーバとを示すブロック図である。

【図4】クライアントユーザインターフェースと、サーバアプリケーションと、セキュリティサーバと、CGI-BINプログラムとのプロセスを示すブロック図である。

【図5】図4に示すクライアントシステムのブラウザプログラムが実行するプロセスを示すフローチャートである。

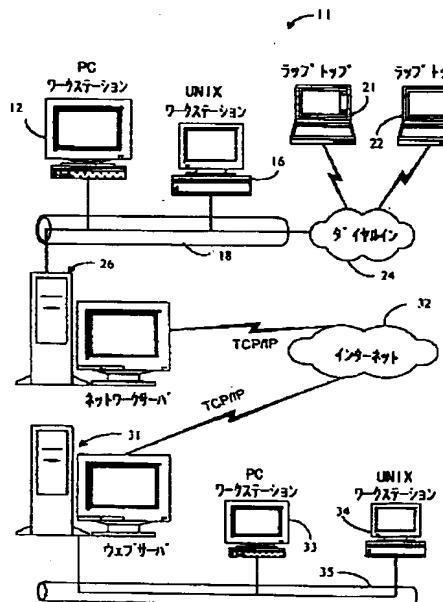
【図6】図4に示すサーバシステムのサーバアプリケーションが実行するプロセスを示すフローチャートである。

【図7】図4に示すサーバシステムのセキュリティサーバプログラムのプロセスを示すフローチャートである。

【図8】図4に示すCGI-BINプログラムプロセスのプロセスを示すフローチャートである。

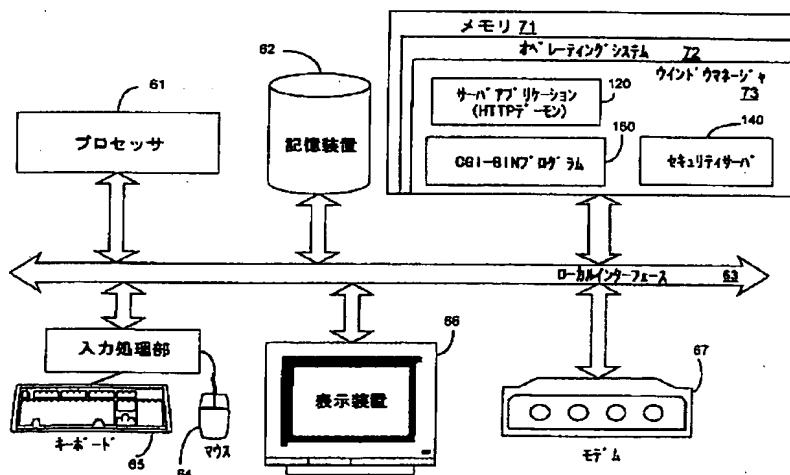
【符号の説明】

【図1】

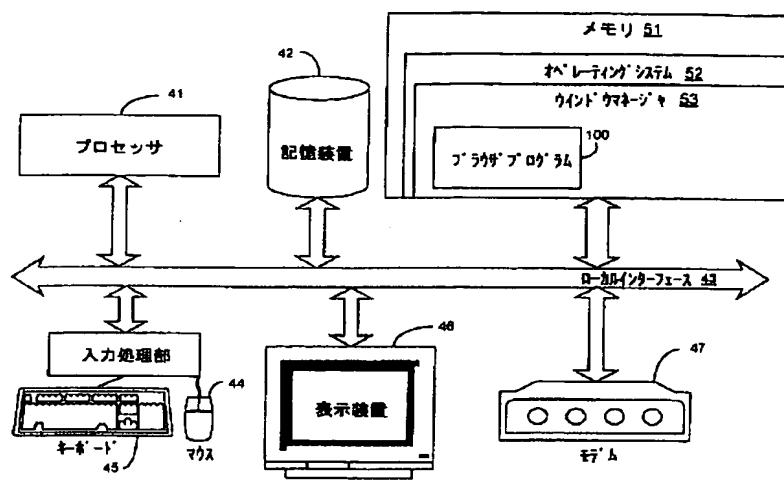


- 11, 18 ネットワーク
- 12, 16, 21, 22, 33, 34 ワークステーション(クライアントシステム)
- 26 ネットワークサーバ(サーバシステム)
- 31 ウェブサーバ(サーバシステム)
- 32 インターネット
- 35 LAN
- 41, 61 プロセッサ
- 42, 62 記憶装置
- 44, 64 マウス
- 45, 65 キーボード
- 46, 66 表示装置
- 47, 67 モデム
- 51, 71 メモリ
- 52, 72 オペレーティングシステム
- 53, 73 ウィンドウマネージャ
- 100 ブラウザプログラム
- 120 サーバアプリケーション
- 140 セキュリティサーバ
- 20 160 CGI-BINプログラム

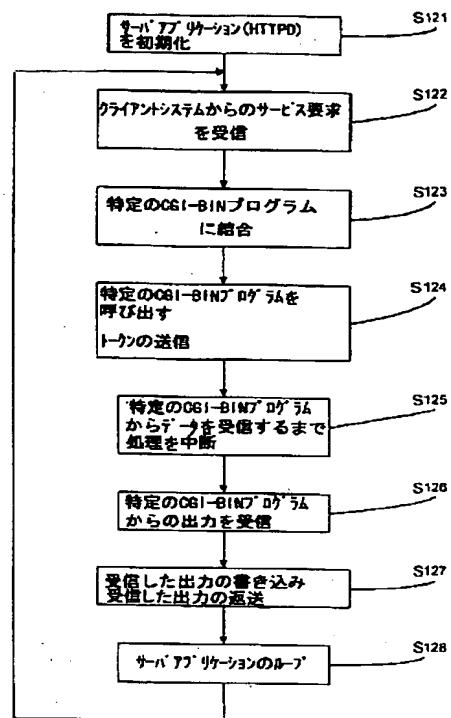
【図3】



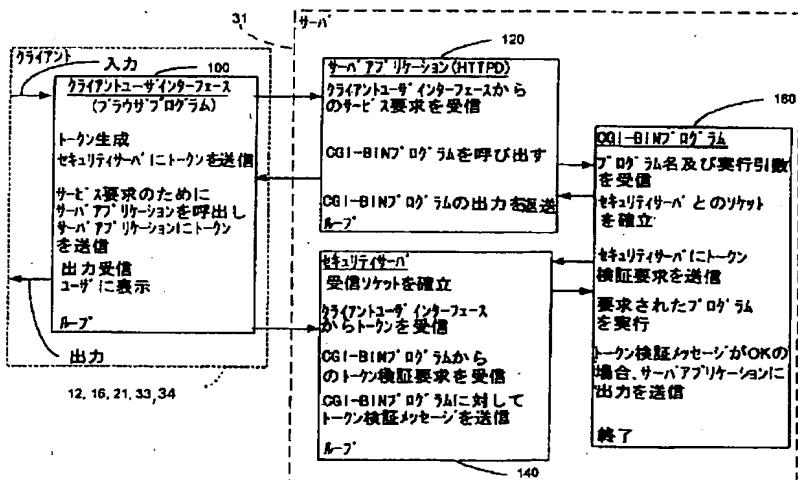
【 図2 】



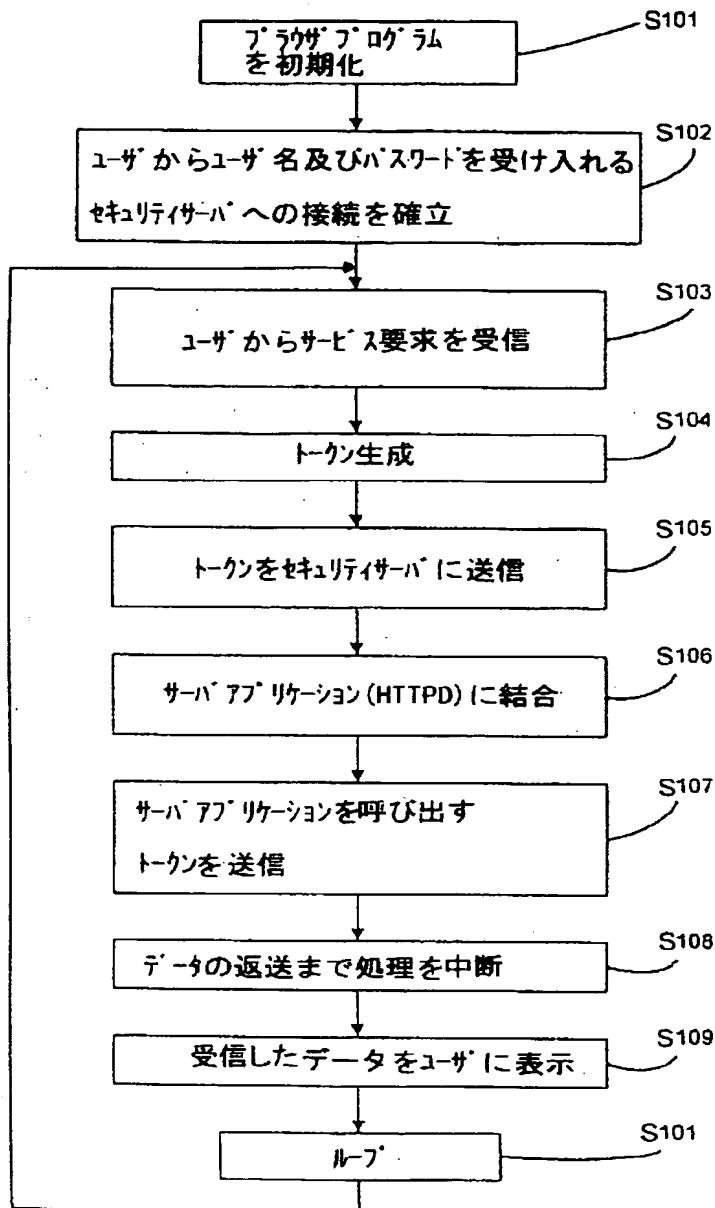
【 図6 】



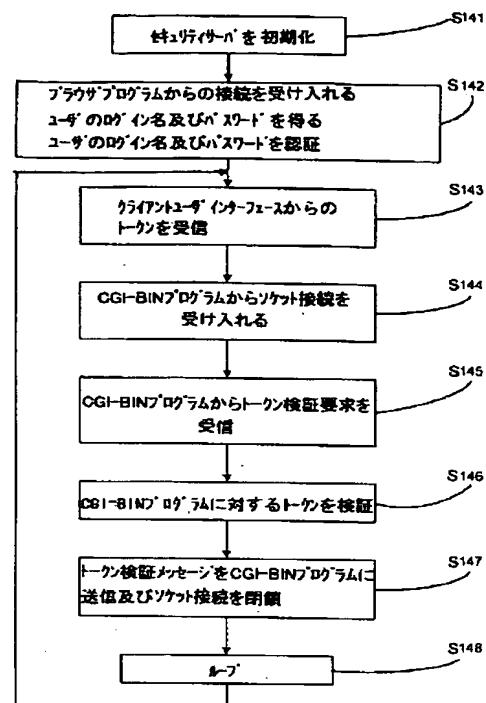
【 図4 】



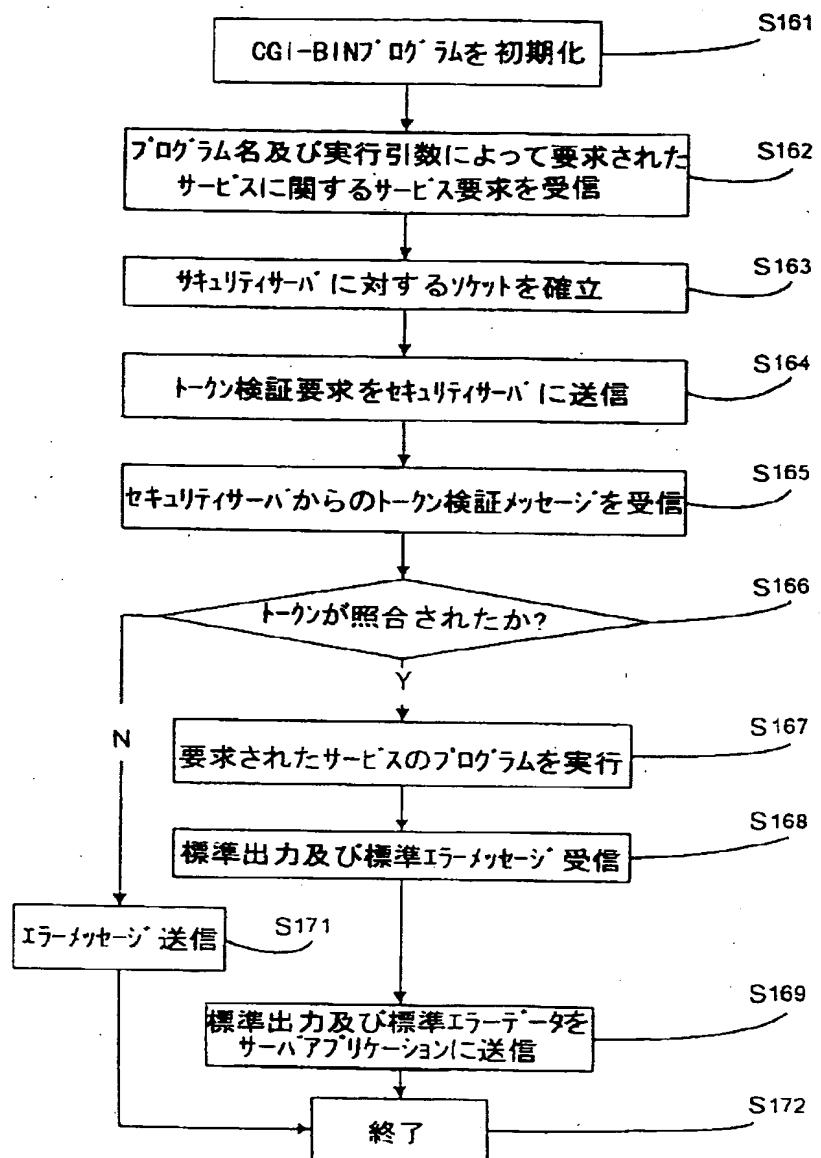
【 図5 】



【 図7 】



【 図8 】



フロントページの続き

(72)発明者 トッド・エム・ゴイン  
 アメリカ合衆国 コロラド, ラブランド,  
 ファイヤソーン ドライブ イースト  
 8320

(72)発明者 クライグ・ダブリュー・ブリアント  
 アメリカ合衆国 コロラド, フォートコリ  
 ンス, ウエリントン・ドライブ 4321